REMARKS

Claim 15 has been amended. Claims 1, 2, 4-11, 13-15, 17, 18, 20-16, 28-31, 33-36, 51, 52, 54-57, 73-78 and 80 are pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Allowed Claims:

Claims 33-36, 75, 51, 52, 54-57, 78 and 80 have been allowed by the Examiner.

Claims Objected To But Otherwise Allowable:

Claims 4-6, 13, 14, 20, 29, 30, 59, 77, 78 and 80 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form.

Claim Objections:

The Examiner objected to claim 15 because it depended from canceled claim 12. Claim 15 has been amended to overcome the Examiner's objection.

The Examiner also objected to claim 4, asserting, "Claim 4 is depended on the undefined claim 77" and that "claim 77 should be mentioned before claim 4 because the claim 4 is depended on claim 77." Applicants respectfully remind the Examiner that "[d]uring prosecution, the order of claims may change and be *in conflict with the requirement that dependent claims refer to a preceding claim*" and that "the numbering of dependent claims and the numbers of preceding claims referred to in dependent claims should be carefully checked when *claims are renumbered upon allowance*" (emphasis added, M.P.E.P. §608.01.n.IV). As such, the Examiner's objection is improper.

Section 103(a) Rejections:

The Examiner rejected claims 1, 2, 7-11, 17, 18 and 21-26 under 35 U.S.C. § 103(a) as being unpatentable over Brandle et al. (U.S. Patent 5,218,699) (hereinafter "Brandle") in view of Monday (U.S. Patent 6,480,860) and further in view of Juster (U.S. Patent 6,202,089), and claims 28, 73 and 74 as being unpatentable over Brandle in view of Monday and Juster and further in view of Anderson, et al. ("Professional XML") (hereinafter "Anderson"). Applicants respectfully traverse these rejections for at least the reasons presented below.

Regarding claim 1, Brandle in view of Monday and Juster fails to teach or suggest **storing the generated results data to a space service in the distributed computing environment, where the space service is separate from the client, and where the space service is accessible as a service by multiple entities other than the client in the distributed computing environment**.

The Examiner relies on Brandle's queue 116 as a space service. However, Brandle clearly describes queue 116 as a local software queue, not as a space service separate from the client. The Examiner also relies on Juster's teachings regarding remote procedure calls, citing col. 5, lines 10-27 and col. 10, line 18 and referring to the Microsoft Message Queue Server (MSMQ) environment, as taught by Juster. However, the Examiner's reliance on Juster and on the combination of Brandle, Monday and Juster is misplaced.

Brandle clearly describes queue 116 as a local software queue, not as a separate service in a distributed computing environment that is accessible by multiple entities other than the client. The Examiner contends that queue 116 "stores generated results data, which provides a queuing service." However, a simple local software queue, such as queue 116, is not a service in a distributed computing environment, as services are understood in the art. No one of ordinary skill in the art would consider Brandle's queue 116 as a *space service in a distributed computing environment*. Additionally, Monday and Juster, whether considered alone or in any combination with Brandle, all fail to teach storing generated results data to space service in a distributed computing environment.

The Examiner argues that Brandle's queue 116 is a space service in the distributed computing environment since the results and information were passed from one service to the other. **However, the actual teachings of the reference do not support the Examiner's interpretation.** Brandle very clearly describes that a local node uses a queue 116 to **locally** store received results "for later retrieval by the application" executing on the same node. Using a local queue to locally store information for retrieval by a local application does not in any way teach or suggest storing the generated results data to a space service in the distributed computing environment, let alone a space service that is separate from the client and accessible as a service by multiple entities other than the client in the distributed computing environment. A local software queue is not such a service in a distributing computing environment.

The Examiner argues that Juster teaches a "space service is separate from said client and wherein said space service is accessible as a service" (Final Action, p. 4). **However, the actual teachings of Juster, even in view of Brandle and Monday, do not support the Examiner's position.** Juster teaches a server providing a plurality of remote procedure call (RPC) service endpoints at runtime on a single server process and providing a dynamic endpoint for responding to endpoint address queries (Juster, col. Abstract, col. 3, lines 37-44). Thus, Juster teaches a system that provides message endpoints and provides a service to respond to endpoint queries. Juster does not describe anything about a "space service [that] is separate from said client" as asserted by the Examiner (Final Action, p. 4). Instead, Juster teaches that a server may receive messages from remote clients querying for RPC endpoints and the server may respond with RPC endpoint addresses so that the clients may send RPC messages to the server. Once the client has and endpoint address, the client may send RPC messages to the server which server applicants may respond to. For example, Juster teaches that in the MSMQ environment, when a server "application receives a request message, it processes the request by reading the contents of the message and acting accordingly" and that "the receiving [server] application can send a response message back to the original requester" (Juster, col. 5, lines 9-19). The actual teachings of Juster do not support the Examiner's position. Juster is silent, even if combined with Brandle and Monday, regarding storing results in a space service or that a space service may be separate from the client, contrary to the Examiner's contention.

As noted above, Juster does not teach storing results to a space service separate from the client. As shown above, Juster teaches that RPC messages are used within the MSMQ environment for clients to send request messages to server applications and that the server applications send return messages to the requesting client. Thus, **Juster <u>explicitly</u> fails to teach** *storing results to a space service that is separate from the client*.

Since both Brandle and Juster fail to teach or suggest storing results to a space service separate from the client, no combination of Brandle and Juster, even in view of Monday, would include such functionality, even if combined with Monday, which also fails to teach or suggest storing results to a space service separate from the client.

Moreover, the RPC teachings of Juster would not cause one to modify Brandle's local software queue 116 or to modify the manner in Brandle's system handles the response data stored in queue 116. Brandle already utilized remote procedure call mechanisms in his system, but chose to utilize a local software queue 116 for storing results data. Thus, Brandle specifically chose not to use his own remote messaging mechanism for storing the results data. Instead, Brandle teaches that "the results returned by the application procedure 118 can returned … for placement into the queue 116, or they can be returned … to the application 100." Thus, Brandle teaches that results are either stored on a local queue or returned to the originating application. Thus, the mere existence of remote procedure calls and a server providing an endpoint querying service, as taught by Juster, is not a valid reason for modifying Brandle's system as suggested by the Examiner, since Brandle's system already includes remote messaging mechanisms that are not used in conjunction with storing results to queue 116.

Additionally, it would not make any sense to modify the system of Brandle to use the remote procedure call mechanisms of Juster to communicate with the local software queue 116. Monday explicitly describes queue 116 as a local software queue. Brandle clearly teaches that queue 116 is part of the client system that originates the remote procedure call, not separate as recited in Applicants' claim (Brandle, FIGs. 1 and 4; column 2, lines 40-45; column 7, lines 41-68).

In further regard to claim 1, Brandle in view of Monday and Juster also fails to teach or suggest **providing an advertisement for the stored results data to the client, where the advertisement comprises information to enable access by the client to the stored results data and the client accessing the stored results data from the space service in accordance with information in the provided advertisement.** The Examiner admits that Brandle fails to teach this limitation and relies upon Monday, citing column 1, lines 50-55 and 59-64, as well as column 9, lines 52-62. The Examiner refers to Monday's teachings regarding a markup language for accessing data in a database. The Examiner argues, "[t]he markup language is preferably defined in extensible markup language (XML) by creating suitable document type definition." However, Monday, even if combined with Brandle and Juster, does not teach or suggest *providing an advertisement for stored results data to the client where the advertisement comprises information to enable access by the client to the stored results data.*

Monday states that a bridge interprets a data request from a client browser in a markup language format and formulates a suitable database query and the resulting data is delivered to the client. (Monday, Abstract; column 1, lines 49-65 and column 10, lines 23-34). Monday further teaches the use of document type definitions (DTDs) that define a grammar for accessing data in a database. Thus, Monday teaches a language and grammar for remotely accessing databases via client browsers. **However, using a markup language and DTDs to allow a user access to a database via a browser is very different from Applicants' claim.** Monday's markup language DTD's are designed to allow a user to formulate queries to gather data from databases. Monday, even if combined with Brandle and Juster, does not teach providing an advertisement for a specific set of results data stored in a space service.

Even if combined with Brandle's and Juster's remote procedure call systems, Monday's markup language does not teach or suggest the specific limitation of *providing an advertisement for stored results data to the client where the advertisement comprises information to enable access by the client to the stored results data.* Instead, a system resulting from the Examiner's combination of Brandle and Monday would perform the remote procedure calls as taught by Brandle and Juster and would also allow a user to access a database via a browser, as taught by Monday.

Moreover, **the Examiner also fails to provide a proper reason to combine Monday with Brandle and Juster.** The Examiner asserts that it would have been obvious to modify Brandle to include the teaching of Monday, "because this allows a user to easily access data in database without knowing a specialized database query language." However, as noted above, Brandle is not concerned at all with providing easy access to data in databases. The Examiner stated reason has absolutely nothing to do with Brandle's system. In fact, the Examiner's stated reason is simply a description of Monday's system. A person seeking to provide easy access to data in databases would simply use Monday's system. There is no reason in the prior art to combine the disparate teachings of Brandle and Monday.

Furthermore, the Examiner is relying on Brandle's local queue 116 for storing the response from a remote procedure call. There is no need, nor benefit, to using Monday's markup language database retrieval system with Brandle's local queue. Local queues are not databases and do not use or require use "specialized database query language" Monday's system is designed to avoid. The Examiner's proposed modification makes no sense. In fact, modifying Brandle's system to use Monday' markup language database access would *not* make it easier for a user to access the responses stored in Brandle's queue 116. Monday makes it clear that this system makes it easy "for a user that has experience with browsers, such as web browsers used to access information via the internet" (Monday, column 4, line 60 – column 5, line 2). A system resulting from the Examiner's combination of Brandle and Monday would still require a user and a browser-based user interface to use Brandle's markup language query system. Such a system requiring a user and a browser is clearly incompatible with Brandle's programmatic remote procedure call system.

One of ordinary skill in the art would have had no reason to include a Monday's browser-type database access system into Brandle's remote procedure call system. To the contrary, modifying Brandle's system to force a user to retrieve the response data from queue 116 using Monday's markup language database access system, even if possible, would greatly complicate Brandle's system and would require user involvement for retrieval of any response data from queue 116. Thus, the Examiner's reasoning for combining Brandle and Monday doesn't make sense. Brandle's system specifically does not employ user intervention to retrieve response data. Instead, Brandle teaches programmatic remote procedure calls that are performed by automatically by application software, not by users (Brandle, Abstract; column 2, lines 20-45; column 3, lines 25-54).

Thus, as shown above, the Examiner's combination of cited art fails to teach or suggest all the limitations of Applicants' claim. As such, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested.

In regards to claim 17, Brandle in view of Monday and Juster fails to teach or suggest **a <u>space service device configured to receive and store results data</u> from service devices in the distributed computing system, wherein the <u>space service device is a separate physical device than the client device</u>**. As shown above regarding the rejection of claim 1, the Examiner's reliance on Juster and on the combination of Brandle, Monday and Juster is misplaced. The Examiner relies on Brandle's queue 116 as a space service. However, Brandle clearly describes queue 116 as a local software queue, not as a space service separate from the client. The Examiner also relies on Juster's teachings regarding remote procedure calls, citing col. 5, lines 10-27 and col. 10, line 18 and referring to the Microsoft Message Queue Server (MSMQ) environment, as taught by Juster.

As noted above, Brandle clearly describes queue 116 as a <u>local software queue</u>, not as a separate service in a distributed computing environment that is accessible by multiple entities other than the client. The Examiner contends that queue 116 "stores generated results data, which provides a queuing service." However, a simple local software queue, such as queue 116 is not a service in a distributed computing environment, as services are understood in the art. One of ordinary skill in the art would not consider Brandle's queue 116 as a *space service in a distributed computing environment*. Additionally, Monday and Juster both fail to teach storing generated results data to space service in a distributed computing environment.

The Examiner argues that Brandle's queue 116 is a space service in the distributed computing environment since the results and information were passed from one service to the other. **However, the actual teachings of the reference do not support the Examiner's interpretation.** Brandle very clearly describes that a local node uses a queue 116 to **locally** store received results "for later retrieval by the application" executing <u>on the same node</u>. As argued above regarding the rejection of claim 1, using a local queue to locally store information for retrieval by a local application does not in any way teach or suggest storing the generated results data to a space <u>service</u> in the <u>distributed</u> computing environment, let alone a space service that is <u>separate from the client</u> and accessible as a service <u>by multiple entities other than the client in the distributed computing environment</u>. A local software queue is not such a service in a distributing computing environment.

Furthermore, Juster teaches a server providing a plurality of remote procedure call (RPC) service endpoints at runtime on a single server process and providing a dynamic endpoint for responding to endpoint address queries (Juster, col. Abstract, col. 3, lines 37-44). Thus, Juster teaches a system that provides message endpoints and provides a service to respond to endpoint queries. As Applicants' argue above regarding claim 1, Juster does not describe anything about a "space service [that] is separate from said client" as asserted by the Examiner (Final Action, p. 4). Instead, Juster teaches that a server may receive messages from remote clients querying for RPC endpoints and the server may respond with RPC endpoint addresses so that the clients may send RPC messages to the server. Once the client has and endpoint address, the client may send RPC messages to the server which server applicants may respond to. For example, Juster teaches that in the MSMQ environment, when a server "application receives a request message, it processes the request by reading the contents of the message and acting accordingly" and that "the receiving [server] application can send a response message back to the original requester" (Juster, col. 5, lines 9-19). The actual teachings of Juster do not support the Examiner's position. Juster is silent, even if combined with Brandle and Monday, regarding storing results in a space service or that a space service may be separate from the client, contrary to the Examiner's contention.

As noted above, Juster does not teach storing results to a space service separate from the client. As shown above, Juster teaches that RPC messages are used within the MSMQ environment for clients to send request messages to server applications and that the server applications send return messages to the requesting client.

Since both Brandle and Juster fail to teach or suggest storing results to a space service separate from the client, no combination of Brandle and Juster, even in view of Monday, would include such functionality, even if combined with Monday, which also fails to teach or suggest storing results to a space service separate from the client.

Moreover, the RPC teachings of Juster would not cause one to modify Brandle's local software queue 116 or to modify the manner in Brandle's system handles the response data stored in queue 116. Brandle already utilized remote procedure call mechanisms in this system, but chose to utilize a local software queue 116 for storing results data. Thus, Brandle specifically chose not to use his own remote messaging mechanism for storing the results data. Instead, Brandle teaches that "the results returned by the application procedure 118 can returned … for placement into the queue 116, or they can be returned … to the application 100." Thus, Brandle teaches that results are either stored on a local queue or returned to the originating application. Thus, the mere existence of remote procedure calls and a server providing an endpoint querying service, as taught by Juster, is not a valid reason for modifying Brandle's system as suggested by the Examiner, since Brandle's system already includes remote messaging mechanisms that are not used in conjunction with storing results to queue 116.

Additionally, it would not make any sense to modify the system of Brandle to use the remote procedure call mechanisms of Juster to communicate with the local software queue 116, thereby making queue 116 remote from the rest of Brandle's system. However, Monday describes queue 116 as a local software queue. Brandle clearly teaches that queue 116 is part of the client system that originates the remote procedure call, not separate as recited in Applicants' claim (Brandle, FIGs. 1 and 4; column 2, lines 40-45; column 7, lines 41-68).

Thus, for at least the reasons above, the rejection of claim 17 is not supported by the cited art and removal thereof is respectfully requested.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-57500/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: January 11, 2008